Lab1 – A review of C++ Objects

You will work alone on this lab.

(0) Download the Starter Kit from our Algorithms Lab onCourse site.

(1) We'll do a short walk-through of the code together.

(2) Set up a C++ Project, load the three files, and run the code. You should get some minimal output.

(3) Alter the overloaded output `operator<<()` so it prints objects out in a different format. Do something (quickly) that makes your output alil' more "fancy".

(4) Add an additional Constructor (CTOR) that takes only the first and last name. The political affiliation and gender are unknown and should receive default values. Add code in `main()` to create another object that uses your new CTOR.

(5) Add code in `main()` to test/use the `FBvisitor` method `getFullName()`. *Note: see my sample code in main() to keep a running test suite of each of your methods. Continue to add to your main() in this same fashion for each new functional addition as you proceed through the lab; at each step, you will re-run all the tests, of course checking to*

(6) Add code in `main()` to test/use the `FBvisitor` overloaded `operator==()`.

(7) Add a suite of getter and setter methods for all private data members. For example, here are the method prototype declarations for the "getter" to return the object's firstname and the associated "setter" to change the object's firstname.

```
string getFirstName() const;
void   setFirstName(string newName);
```

In the `FBvisitor.h` file, keep the getter prototype declarations all in one block and in the `FBvisitor.cpp` (implementation/definition) file all the method definitions together in one block.

(8) Add a two new private members to the object declaration: (i) an age (integer) and (ii) a GPA. Obviously the age is an integer, but use the integer type `short` to save member.

(9) Update all your other methods to incorporate your two new private members. Re-test all methods.

(10) Add a method `FBvisitor::makeMeRandom()` to randomly generate a user profile. For simplicity sake, you might want to make some static arrays at the start of this method and then your random numbers can be used to pick from the possible values.

```
static const short MAX_MEN   = 2;
static const short MAX_WOMEN = 2;
static string male_fNames[]   = {"Joe", "Pete"};
static string female_fNames[] = {"Sally", "Mary"};
static string lNames[]        = {"Smith", "Jones", "Cnalbel", "Foo"};
static string poli[]          = {"Liberal", "Conservative", "Moderate"};
static string sex[]           = {"Male", "Female"};
```